Engineer

Full introduction

Engineer ApS in full: services with examples of delivered work, AI capabilities, platforms, working principles and company details in one place.

Read the complete presentation below: our services and the work behind them, AI capabilities, platforms, working principles and company details. The index also leads to each subject and our Copenhagen workspace.

Explore every subject

- About us — Engineer ApS is a Copenhagen software and IT consultancy, established 2025.
- Portfolio — Browse the full portfolio of Engineer ApS engineering achievements, each with a detailed review.
- Services — What Engineer ApS does — a catalogue of software, data, cloud and IT services, each backed by real project work.
- Tech stack — The technologies Engineer ApS builds with: languages, data platforms, cloud and infrastructure, and how we decide what to use.
- Ready for AI agents — Website and application reviews for crawler access, readable content, structured data and agent interfaces, with scoped implementation.
- Company data ready for AI agents — Connect company records to an agreed AI workflow, with clear data sources, quality checks, permissions and a traceable first result.
- Custom AI agents built for your trade — Custom AI agents for an agreed workflow, connected to your systems and evaluated on your cases, with human review and scoped access.
- Security for the AI agents you run — Access reviews and controls for AI agents: an agreed inventory, prioritized findings, permission tests and a verification record.
- Your software as tools for AI agents — Expose selected software operations as described tools for AI agents, with permissions, contract checks and a planned rollout.

- Our working setup in Copenhagen

We build the systems **your company runs on** — and we stand behind them.

Software, data, cloud and the IT that keeps it all running — from Copenhagen, since 2025.

---

company
Engineer
Established 2025

Software Development & IT Consulting Services

+45 50 10 61 84 · Copenhagen, Denmark

Carefully Crafted With Love by D.

We work with **the platforms** your business already runs on.

Apple Microsoft DigitalOcean AWS Google

See the whole stack

Ready for the AI agents your customers send

Found

A robots file that says what agents may do, a sitemap with true dates, and every change announced through IndexNow the day it ships.

Read

A clean Markdown copy of every page, an llms.txt map for language models, and structured data that says what you do.

Usable

Accessible forms and controls, plus a described API for the application operations selected for agent access.

See how it works Ask about your site →

Company data your AI agents can answer from

Gathered

Your CRM, ERP, shared drives and mail archives brought into one database an agent reaches through one described door.

Cleaned

Agreed identifiers and quality checks connect the selected data sources, with duplicate handling and validation rules documented.

Permissioned

Access checks and audit logging follow the permissions and data boundaries agreed for the integration.

See the data work Ask about your data →

AI agents built for the work your trade does

Connected

Wired to your ERP, CRM, booking system and mailbox through described tools, each with one narrow job.

Measured

Scored against real cases your experts write, so every change to a prompt, a tool or a model is tested before it ships.

Supervised

What it does alone and what a person approves, written down with you, enforced in the tools and traced step by step.

See how agents are built Ask about your trade →

Security review for the AI agents in scope

Mapped

An agreed inventory of the agents, credentials and data access in scope, kept with your configuration.

Locked

Each agent on its own identity and least privilege, and the text it reads treated as data, never as orders.

Recorded

Every action logged with the person it was for, where the agent cannot change it, and alerts on the unusual.

See how agents are locked Ask about your agents →

The software you run, made into tools agents can call

Unpicked

The rules hidden in screens, forms and stored procedures found, written down and moved into one layer.

Exposed

Each operation offered as a typed, described tool over an API or an MCP server, acting only within its user's rights.

Kept running

A scoped integration can expose one operation at a time alongside existing screens; application changes are agreed after reviewing its interfaces.

See how software opens up Ask about your software →

Read the work **in its context** — the decisions, delivery and reported outcomes.

See the whole catalogue

One partner for **38 services** across six disciplines.

Platform and backend, data and databases, cloud and DevOps, consulting and leadership, IT support, brand and marketing.

Browse the catalogue

{ Software Engineering }

Deciding how the whole thing should be built

Platform & Solution Architecture

The shape of a system worked out on paper first, so the parts fit and the cost is known before anyone writes code.

Adopted Swift 6 strict concurrency over C

Adopted Swift 6 complete strict concurrency with no actors, bridging a blocking C event loop to the main actor through one producer, one consumer and one ordering — after establishing that a task per event loses the ordering the interface depends on.

A bounded event pipeline for a macOS app

Bounded event buffers and logging after a recorded macOS incident with 41 MB per second growth and 111 GB of compressed pages on a 36 GB machine; a recorded full test run reduced core log lines from 610,996 to 1,411.

Built a native macOS client in Swift 6

Built a native macOS messaging client in Swift 6 and SwiftUI — 11,141 lines across 53 files — over the C interface of a Rust core linked as a static archive from a pinned revision.

See all 20 achievements

How we work

Understand first

We learn the problem before we propose a solution. The brief you arrive with is the start of that conversation, not the end of it.

Build it to last

Durable over quick, clarity over cleverness. The system still has to be workable by the people who inherit it.

Stand behind it

We agree any ongoing operation or support as part of the engagement. Handover makes responsibilities and the next contact route clear.

{ Software Engineering }

The engine behind your app or website

Backend & API Development

The part users never see: where the data lives, what the rules are, and how other systems are allowed to talk to it.

Reached the unused half of the core

Reached the half of the messaging core the application had never used — backup transfer, disappearing messages, message editing and resending, verified invitations, proxies and encryption policy — driving every test against the real library with no mocks.

Wrote a parser verifying every FFI call

Wrote a parser that reads the real 7,308-line C header and verifies every call site, every enum constant and that every pointer-owning class is final, after a hand-written placeholder header let calls to three removed functions compile, link and crash.

Adopted Swift 6 strict concurrency over C

Adopted Swift 6 complete strict concurrency with no actors, bridging a blocking C event loop to the main actor through one producer, one consumer and one ordering — after establishing that a task per event loses the ordering the interface depends on.

See all 32 achievements

{ Software Engineering }

The screens people actually use

Frontend Development

Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Built a 48-token macOS design system

Built a 48-token design system on the platform's own glass material and wrote the linter that rejects a magic number, a hardcoded font size, or an animation that ignores the reduce-motion preference.

Named all 17 icons for screen readers

Named every icon-only control in the interface for screen readers after finding the send button announced as "arrow up circle, button", and wrote the linter that requires the label within eight lines of the icon.

A bounded event pipeline for a macOS app

Bounded event buffers and logging after a recorded macOS incident with 41 MB per second growth and 111 GB of compressed pages on a 36 GB machine; a recorded full test run reduced core log lines from 610,996 to 1,411.

See all 30 achievements

{ Software Engineering }

A working product, end to end

Full-Stack Product Development

One team taking an idea from a sketch to something your customers can log into, instead of handing it between three suppliers.

Built a native macOS client in Swift 6

Built a native macOS messaging client in Swift 6 and SwiftUI — 11,141 lines across 53 files — over the C interface of a Rust core linked as a static archive from a pinned revision.

Built a database-driven CV generator

Built a database-driven CV, references, portfolio and cover-letter generator in Python — 41 modules, 10,580 lines — rendering six output formats from one 23-table SQLite source assembled by a 19-step idempotent pipeline.

Built a multilingual Hugo site in 3 languages

Built a multilingual static site in Hugo across 108 template files including 53 partials, publishing every page in four representations from one content tree in three languages, and again under seven focused subdomains built from that same tree.

See all 16 achievements

{ Software Engineering }

A website you can update yourself

Website Development & CMS

A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

Built the visual identity from 2 drawings

Built the company's icon and favicon sets from two hand-made drawings, with every derived asset regenerated by script and a check that fails when a derived file was committed before its source.

Passed 106 of 106 PDF/UA-1 profile checks

Recorded PDF/UA-1 validation across nine documents with 106 of 106 configured accessibility-profile checks passing, while the archival variant failed one of 146 rules.

See all 22 achievements

{ Software Engineering }

Showing up when customers ask an AI

AI Agent Readiness

A site or an app that the AI assistants your customers use can find, read and act on, checked on every release rather than promised once.

Ran development as 51 written initiatives

Ran the site's development as 51 written initiatives across 54 plan documents, each carrying its open boxes, its measurements and the decisions it declined — including six of eight agent-readiness findings refused with the reason recorded.

Fixed a sitemap with one shared timestamp

Fixed a sitemap where 172 of 176 URLs shared one modification timestamp, by taking the date from git history after establishing that the export rewrites every file on every run.

Replaced 963 metadata blocks with one graph

Replaced 963 anonymous structured-data blocks that restated the company 1,671 times with one linked graph of 16 types minted from stable origin identifiers.

See all 5 achievements

{ Software Engineering }

Letting AI use the software you already have

AI Agent Tools & MCP Servers

Your existing software opened up as tools an AI agent can call, without rewriting anything that already works.

Built the operator back-office for the platform

Built the operator back-office — around 40 admin routes and 128 components covering claims, moderation, feature flags, cache and diagnostics — so the platform can be run without database access.

Moved slow work onto a River job queue

Moved slow work off the request path onto a River job queue — 15 worker modules, 8 scheduled tasks and 20 pg_cron jobs — so a request returns while the work behind it carries on.

Generated the API contract from the database out

Generated the API contract outward from the database — OpenAPI, a 44,076-line typed TypeScript client, 61 mock handlers and the limits the UI enforces — with a guard at every hop that fails on drift.

See all 5 achievements

{ Software Engineering }

Maps that answer questions

GIS & Geospatial Solutions

Anything that happens in a place — vessels, vehicles, plots, sensors — put on a map you can search, measure and share.

A mapping product, carried end to end

Architected, developed and supported infrastructure, data processing and a map application over 2 years, covering both software delivery and operational maintenance.

Led 30+ GIS projects from build to support

Led the development, deployment, and support of over 30 GIS projects, demonstrating expertise in PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS, and Mapbox technologies.

Managed 30 successful GIS projects to delivery

Managed the execution of 30 successful GIS projects, demonstrating leadership in delivering innovative solutions across the field.

See all 13 achievements

{ Software Engineering }

Making it obvious how to use

UI/UX Design & Design Systems

Layout, wording and flow arranged so people finish what they came to do instead of ringing your support line.

Built the visual identity from 2 drawings

Built the company's icon and favicon sets from two hand-made drawings, with every derived asset regenerated by script and a check that fails when a derived file was committed before its source.

Rewrote error messages to stop blaming users

Rewrote the application's error messages against a written tone standard after a refused sign-in blamed the user for mistyping when the provider actually required an app-specific password, and covered it with a test that names the provider.

Built a 48-token macOS design system

Built a 48-token design system on the platform's own glass material and wrote the linter that rejects a magic number, a hardcoded font size, or an animation that ignores the reduce-motion preference.

See all 20 achievements

{ Software Engineering }

Working in more than one language

Internationalization & Localization

Software that reads naturally in every market you sell in, including the dates, the currencies and the way names sort.

Built a cross-language figure check

Built a cross-language content check that fails when a translation drops a figure the English states, and when a language uses notation it does not use — finding two Danish descriptions missing a metric and sixteen Ukrainian spans quoting in the English style.

Moved every string into a content tree

Moved every user-facing string out of Python into a content tree of 874 files across three languages, after finding dead translations nobody could see were dead and a check silently grading a third of the achievements.

Built a database-driven CV generator

Built a database-driven CV, references, portfolio and cover-letter generator in Python — 41 modules, 10,580 lines — rendering six output formats from one 23-table SQLite source assembled by a 19-step idempotent pipeline.

See all 8 achievements

{ Software Engineering }

An AI assistant that knows your trade

Custom AI Agent Development

An agent wired to the systems you use, tested on cases your own experts write, overseen by your people and never tied to one model.

Made six collectors state what they skipped

Made six reporting collectors state the part of their input they never read — files that stayed shut, probes not installed, journal lines unrecognised — and fitted thresholds to 116 measured samples, after establishing that a blind collection and a healthy host render the same page.

Wrote tests for the checkers themselves

Wrote tests for the checkers themselves after establishing that a checker fed only clean input will one day report clean because it read nothing — planting a misspelling to confirm the spell-check finds it, and taking an id range from the database rather than from a number in the test.

Built a 22-linter commit gate

Built a commit gate of 22 one-line linters plus five that earn a paragraph, with no warning tier and no inline suppressions permitted, covering HTML, CSS, JavaScript, Python, YAML, Markdown, shell, links, spelling, secrets and typography.

See all 5 achievements

{ Data & Databases }

Somewhere sensible to keep your data

Database Design & Modeling

A structure that matches how the business actually works, so the questions you need answered later are still possible to ask.

Built a database-driven CV generator

Built a database-driven CV, references, portfolio and cover-letter generator in Python — 41 modules, 10,580 lines — rendering six output formats from one 23-table SQLite source assembled by a 19-step idempotent pipeline.

Built the hiring and seafarer career workspaces

Built hiring and seafarer career workspaces; the recorded implementation snapshot contains 151 stored functions across 48 tables for vacancies, applications, certificates, rank progression and sea time.

Built a loyalty wallet, leagues and badges

Built the loyalty and reputation system — 67 functions over a 31-table ledger, with leagues, badges and a redemption shop — taking a row lock on the balance to close the double-spend window.

See all 14 achievements

{ Data & Databases }

Looking after the database

Database Administration (DBA)

Somebody responsible for the thing the whole company rests on: it is up, it is backed up, and it is not quietly rotting.

1,022 migrations behind a schema drift gate

Kept the schema honest across 1,022 migrations with a CI gate that builds the database both ways — a fresh install, and an install plus every migration — and fails when the two disagree.

Implemented backups and disaster recovery as code

Configured database backup retention as infrastructure-as-code, then audited the recovery position and documented the restore procedure — naming the remaining gaps rather than leaving them to be found during an incident.

Engineered 600 PL/pgSQL ETL/ELT pipelines

Engineered 600 PL/pgSQL-based ETL/ELT pipelines to streamline complex data processing workflows across multiple PostgreSQL development and production environments.

See all 9 achievements

{ Data & Databases }

Making the slow parts fast

Database Performance Tuning

Finding the queries and the missing indexes behind a system that has got heavier every year, and fixing the ones that matter.

Time-ordered keys for a growing data model

Adopted UUID v7 time-ordered identifiers (PostgreSQL 18) as entity keys to reduce B-tree index fragmentation and speed up queries.

Accelerated PostgreSQL performance by 10x

Accelerated PostgreSQL performance by 10x via strategic indexing, partitioning, and query optimization, enhancing database efficiency for user, tenant, geospatial, and time-series electrical data.

Ran 100 PostgreSQL, MS SQL, BigQuery warehouses

Architected, created, and managed 100 PostgreSQL, MS SQL, and Google BigQuery data warehouse databases with primarily GIS and time-series data, optimizing performance and scalability.

See all 6 achievements

{ Data & Databases }

Getting off the old system

Database Migration & Modernization

Data taken out of something ageing and into something supported, with the business still running while it happens.

Moved every string into a content tree

Moved every user-facing string out of Python into a content tree of 874 files across three languages, after finding dead translations nobody could see were dead and a check silently grading a third of the achievements.

1,022 migrations behind a schema drift gate

Kept the schema honest across 1,022 migrations with a CI gate that builds the database both ways — a fresh install, and an install plus every migration — and fails when the two disagree.

Improved GIS map performance 10x on PostgreSQL

Improved geographical map application performance by 10x through strategic database transition from MSSQL to PostgreSQL, including the data cutover, optimizing processing and data security.

{ Data & Databases }

Getting your data from A to B on its own

Data Pipeline Development (ETL/ELT)

Numbers arriving where they are needed, on a schedule,
without anyone copying a spreadsheet at the end of the month.

Generated 495 pages from a SQLite export

Generated 495 achievement pages across three languages from a read-only SQLite export, with the page address authored as data so that correcting a sentence no longer moved the page and broke the link.

Built first-party analytics inside PostgreSQL

Built first-party product analytics in PostgreSQL — 47 functions over partitioned event tables that prune themselves — pseudonymised behind a rotating salt and gated on the visitor's consent.

A mapping product, carried end to end

Architected, developed and supported infrastructure, data processing and a map application over 2 years, covering both software delivery and operational maintenance.

See all 14 achievements

{ Data & Databases }

One place where all the numbers agree

Data Warehouse Design

Sales, operations and finance figures brought together, so two departments stop bringing two answers to the same meeting.

Ran 100 PostgreSQL, MS SQL, BigQuery warehouses

Architected, created, and managed 100 PostgreSQL, MS SQL, and Google BigQuery data warehouse databases with primarily GIS and time-series data, optimizing performance and scalability.

Designed and ran 6 ETL/ELT pipelines

Designed, implemented, and administered 6 ETL/ELT pipelines, utilizing Google BigQuery, MSSQL, PostgreSQL, Shell scripting, PL/pgSQL, and Transact-SQL, integrating data for efficient Python API processing.

Made a geographical data pipeline 50x faster

Recorded a 50x geographical data-pipeline improvement in the founder's Utiligize work through SQL and data-model changes across PostgreSQL, MS SQL and Google Cloud BigQuery.

{ Data & Databases }

Seeing what is going on

Data Analytics & BI Dashboards

Dashboards and reports showing the handful of figures a decision actually turns on, refreshed without being asked.

Built first-party analytics inside PostgreSQL

Built first-party product analytics in PostgreSQL — 47 functions over partitioned event tables that prune themselves — pseudonymised behind a rotating salt and gated on the visitor's consent.

Designed a time-management and reporting system

Designed a time-management and reporting system that remained in production use for years without significant modification.

Delivered analysis and regular progress reports

Delivered analysis and regular progress reports to the CEO, translating engineering KPIs and delivery status into decisions.

See all 11 achievements

{ Data & Databases }

Trusting the numbers

Data Governance & Quality

Rules and checks that catch bad data on the way in, and a clear answer to who owns which figure and where it came from.

Measured 13 addresses with no tracking

Measured thirteen addresses from the web server's own access log alone — no cookie, no tracker and no third-party service — masking the client network as each line is written, and folding the result into a permanent archive of 21 daily counts and 20 monthly facets.

Know the licence of everything you ship

Audited 644 Rust crates for licence compatibility on every build, and proved the check fires by rewriting one crate's licence and by moving the pinned core revision without regenerating.

Built a cross-language figure check

Built a cross-language content check that fails when a translation drops a figure the English states, and when a language uses notation it does not use — finding two Danish descriptions missing a metric and sixteen Ukrainian spans quoting in the English style.

See all 16 achievements

{ Data & Databases }

Data an AI can actually use

AI Data Readiness

Your records where AI agents can reach them, one version of each customer, nobody shown more than they should, and every answer traced to its source.

Replaced 963 metadata blocks with one graph

Replaced 963 anonymous structured-data blocks that restated the company 1,671 times with one linked graph of 16 types minted from stable origin identifiers.

Generated 495 pages from a SQLite export

Generated 495 achievement pages across three languages from a read-only SQLite export, with the page address authored as data so that correcting a sentence no longer moved the page and broke the link.

1,022 migrations behind a schema drift gate

Kept the schema honest across 1,022 migrations with a CI gate that builds the database both ways — a fresh install, and an install plus every migration — and fails when the two disagree.

See all 6 achievements

{ Cloud, Infrastructure & DevOps }

Where the software runs

Cloud Infrastructure & Migration

Servers sized for what you actually use, moved into the cloud or back out of it, with the monthly bill worked out first.

Provisioned a second server from code

Automated the provisioning of a second server on a second cloud provider, creating the firewall before the machine so it is born behind one, with both providers' firewalls written directly against their REST APIs to avoid a third-party collection.

Ran containers on Podman and Quadlet

Deployed a container plane on Podman and Quadlet under systemd rather than Docker, because Docker's default bridge-port publishing can bypass UFW rules — and gave deploys an unprivileged user with one fixed command instead of root.

Public services on a small server

Hosted the company's public services on a single-core 512 MB host — a git forge, a web server for seven domains, Tor, two alternative-protocol servers and intrusion banning — treating 464 MB of usable memory as the design constraint; backup automation is a separate implementation.

See all 9 achievements

{ Cloud, Infrastructure & DevOps }

Servers you can rebuild from versioned configuration

Infrastructure as Code

The whole setup written down as files, so it can be recreated exactly and never lives only in one person's head.

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

Rehearsed the forge CI hook, found 2 defects

Rehearsed the forge-side CI hook and found two defects unreachable by reading the file: a fallback that put an unresolvable argument on the hook's input, and git's own environment variable following the gate into the checkout and turning 19 tests red.

Split every operational secret per host

Split the four per-host operational secrets after establishing that two hosts sharing one dead-man's switch alert less than two switches, not more, and that a shared backup passphrase can expose both hosts' backups, even in separate repositories, if someone obtains it and can read both repositories' encrypted data and key files.

See all 17 achievements

{ Cloud, Infrastructure & DevOps }

Shipping changes without drama

DevOps & CI/CD Automation

Tested, released and reversible, so changes reach customers on a Tuesday afternoon rather than late on a Friday night.

Held 4 repositories to one history standard

Held four repositories to one history standard — conventional, emoji-free, no attribution trailers, enforced by a commit-message hook — alongside 48 instruction documents that govern how the work is done.

Know the licence of everything you ship

Audited 644 Rust crates for licence compatibility on every build, and proved the check fires by rewriting one crate's licence and by moving the pinned core revision without regenerating.

Reached the unused half of the core

Reached the half of the messaging core the application had never used — backup transfer, disappearing messages, message editing and resending, verified invitations, proxies and encryption policy — driving every test against the real library with no mocks.

See all 43 achievements

{ Cloud, Infrastructure & DevOps }

Software that runs the same everywhere

Containerization & Orchestration

Packaged to behave identically on a laptop, on a test server and in production, and to scale up when a busy week arrives.

Ran containers on Podman and Quadlet

Deployed a container plane on Podman and Quadlet under systemd rather than Docker, because Docker's default bridge-port publishing can bypass UFW rules — and gave deploys an unprivileged user with one fixed command instead of root.

Cut systemd sandbox exposure on every unit

Cut systemd sandbox exposure across every unit this platform installs — a dead-man's-switch service from 9.6 UNSAFE to 1.5, an internet-facing git forge from 8.3 EXPOSED to 1.5 — and added a converge-time parser check after finding a misspelled directive silently ignored in three unit templates.

Built GitHub Actions CI/CD pipelines

Built GitHub Actions CI/CD pipelines with a distroless production frontend image and multi-environment promotion.

See all 5 achievements

{ Cloud, Infrastructure & DevOps }

Knowing before your customers do

Site Reliability & Monitoring

Monitoring and alerts that speak up while a fault is still small, and the figures that show how often it stays that way.

Made six collectors state what they skipped

Made six reporting collectors state the part of their input they never read — files that stayed shut, probes not installed, journal lines unrecognised — and fitted thresholds to 116 measured samples, after establishing that a blind collection and a healthy host render the same page.

Measured 13 addresses with no tracking

Measured thirteen addresses from the web server's own access log alone — no cookie, no tracker and no third-party service — masking the client network as each line is written, and folding the result into a permanent archive of 21 daily counts and 20 monthly facets.

A bounded event pipeline for a macOS app

Bounded event buffers and logging after a recorded macOS incident with 41 MB per second growth and 111 GB of compressed pages on a 36 GB machine; a recorded full test run reduced core log lines from 610,996 to 1,411.

See all 23 achievements

{ Cloud, Infrastructure & DevOps }

Getting everything back after a bad day

Backup & Disaster Recovery

Backups that are tested rather than assumed, and a written plan for the morning a server does not come back on.

Split every operational secret per host

Split the four per-host operational secrets after establishing that two hosts sharing one dead-man's switch alert less than two switches, not more, and that a shared backup passphrase can expose both hosts' backups, even in separate repositories, if someone obtains it and can read both repositories' encrypted data and key files.

Backup automation with an explicit restore method

Implemented restic backup automation with retention, integrity checks and a monthly restore-test workflow; documented storage activation and live-restore verification as separate operational steps.

Implemented backups and disaster recovery as code

Configured database backup retention as infrastructure-as-code, then audited the recovery position and documented the restore procedure — naming the remaining gaps rather than leaving them to be found during an incident.

See all 5 achievements

{ Cloud, Infrastructure & DevOps }

Keeping the lights on

System Administration

Day-to-day care of servers, mail, accounts and updates, so the things everybody relies on stay boring.

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

Made six collectors state what they skipped

Made six reporting collectors state the part of their input they never read — files that stayed shut, probes not installed, journal lines unrecognised — and fitted thresholds to 116 measured samples, after establishing that a blind collection and a healthy host render the same page.

Rehearsed the forge CI hook, found 2 defects

Rehearsed the forge-side CI hook and found two defects unreachable by reading the file: a fallback that put an unresolvable argument on the hook's input, and git's own environment variable following the gate into the checkout and turning 19 tests red.

See all 28 achievements

{ Cloud, Infrastructure & DevOps }

Connecting your offices and your people

Networking & VPN Setup

Wi-Fi, firewalls and remote access set up so staff can work from anywhere and only the right people get in.

Provisioned a second server from code

Automated the provisioning of a second server on a second cloud provider, creating the firewall before the machine so it is born behind one, with both providers' firewalls written directly against their REST APIs to avoid a third-party collection.

Reconciled a 20-record DNS zone as code

Reconciled a 20-record DNS zone declaratively against the Cloudflare API with separate audit and BIND-export entry points, and turned the CDN proxy back off on privacy grounds after building it.

Verified firewall rules by position

Verified firewall rules by position rather than presence, reading the numbered rule list and the live packet-filter chain, because a rule that exists is not a rule any packet reaches.

See all 11 achievements

{ Cloud, Infrastructure & DevOps }

Deciding who can reach what

Security & Access Management

Accounts, passwords and permissions arranged so people hold exactly the access their job needs, and leavers lose it that day.

Published a readable Tor onion mirror

Published the site as a Tor onion mirror on a readable address beginning engineer, with the key generated off the host so it never reached this repository or a laptop, and left it the one front door of four that is never counted.

Measured 13 addresses with no tracking

Measured thirteen addresses from the web server's own access log alone — no cookie, no tracker and no third-party service — masking the client network as each line is written, and folding the result into a permanent archive of 21 daily counts and 20 monthly facets.

Know the licence of everything you ship

Audited 644 Rust crates for licence compatibility on every build, and proved the check fires by rewriting one crate's licence and by moving the pinned core revision without regenerating.

See all 31 achievements

{ Cloud, Infrastructure & DevOps }

Making sure your AI only touches what it should

AI Agent Security

Every agent's access mapped and held to what it needs, every action logged where the agent cannot alter it, and every safeguard tested rather than trusted.

Split every operational secret per host

Split the four per-host operational secrets after establishing that two hosts sharing one dead-man's switch alert less than two switches, not more, and that a shared backup passphrase can expose both hosts' backups, even in separate repositories, if someone obtains it and can read both repositories' encrypted data and key files.

Wrote a scope rule after a credential leak

Wrote a scope rule into the repository after a restructure carried another company's inventory, firewall allowances and prose into it — and kept the quarantined residue under the secret scanner rather than excluding it.

Added a one-second preflight for secrets

Added a preflight play that runs the same code as the converge against operator-local secrets in about a second, after a half-applied production run died on its ninth task with the swap settings already written to the live host.

See all 9 achievements

{ Consulting & Leadership }

A senior engineer in the room

Technical Leadership & Consulting

Somebody on your side of the table when a supplier, a budget or an architecture decision needs a second informed opinion.

Held 4 repositories to one history standard

Held four repositories to one history standard — conventional, emoji-free, no attribution trailers, enforced by a commit-message hook — alongside 48 instruction documents that govern how the work is done.

Took the test suite to 8.9 seconds

Took the test suite from six tests over a sixty-second limit to 135 passing in 8.9 seconds by profiling the main thread and removing the two calls it sat inside for 3,989 samples out of 4,017.

Adopted Swift 6 strict concurrency over C

Adopted Swift 6 complete strict concurrency with no actors, bridging a blocking C event loop to the main actor through one producer, one consumer and one ordering — after establishing that a task per event loses the ordering the interface depends on.

See all 26 achievements

{ Consulting & Leadership }

Growing the team you already have

Team Building & Mentoring

Hiring, onboarding and coaching, so the knowledge settles with your own people rather than leaving with a consultant.

Managed a team delivering IT support

Managed a team delivering IT support, data recovery, and hardware repair services to over 1,000 clients, ensuring high-quality service.

Saved 150 hours a month across 30 projects

Enhanced project efficiency, saving 150 hours per month across 30 projects by optimizing workflows and resource management.

Established a Technical Support department

Established a Technical Support department, servicing over 10,000 clients with IT support and troubleshooting solutions, tracking client requests in a CRM.

See all 9 achievements

{ Consulting & Leadership }

Getting it delivered

Project Management (Agile)

Somebody accountable for scope, dates and the awkward conversations, so a project has one owner instead of five opinions.

Held 4 repositories to one history standard

Held four repositories to one history standard — conventional, emoji-free, no attribution trailers, enforced by a commit-message hook — alongside 48 instruction documents that govern how the work is done.

Put the documentation under its own linter

Put the documentation under its own linter — a per-file line budget that only ratchets down, an index requirement and a real-path check — which found six of nine plans marked done while carrying unchecked boxes, and fourteen commit-gating tasks named in no document.

Ran development as 51 written initiatives

Ran the site's development as 51 written initiatives across 54 plan documents, each carrying its open boxes, its measurements and the decisions it declined — including six of eight agent-readiness findings refused with the reason recorded.

See all 16 achievements

{ Consulting & Leadership }

Working out what to build

Product Strategy & Requirements

Turning what the business wants into something a developer can build, and saying out loud what is not worth building.

Rewrote error messages to stop blaming users

Rewrote the application's error messages against a written tone standard after a refused sign-in blamed the user for mistyping when the provider actually required an app-specific password, and covered it with a test that names the provider.

Ran development as 51 written initiatives

Ran the site's development as 51 written initiatives across 54 plan documents, each carrying its open boxes, its measurements and the decisions it declined — including six of eight agent-readiness findings refused with the reason recorded.

Built a 22-linter commit gate

Built a commit gate of 22 one-line linters plus five that earn a paragraph, with no warning tier and no inline suppressions permitted, covering HTML, CSS, JavaScript, Python, YAML, Markdown, shell, links, spelling, secrets and typography.

See all 19 achievements

{ Consulting & Leadership }

Writing it down so the next person can follow

Technical Documentation

Handovers, runbooks and manuals that let your own staff operate the system without ringing whoever built it.

Made six collectors state what they skipped

Made six reporting collectors state the part of their input they never read — files that stayed shut, probes not installed, journal lines unrecognised — and fitted thresholds to 116 measured samples, after establishing that a blind collection and a healthy host render the same page.

Held 4 repositories to one history standard

Held four repositories to one history standard — conventional, emoji-free, no attribution trailers, enforced by a commit-message hook — alongside 48 instruction documents that govern how the work is done.

Rewrote error messages to stop blaming users

Rewrote the application's error messages against a written tone standard after a refused sign-in blamed the user for mistyping when the provider actually required an app-specific password, and covered it with a test that names the provider.

See all 19 achievements

{ IT Support & Hardware }

Everyday computer help

IT Support & Helpdesk

The laptop, the printer, the login and the mailbox, sorted quickly and in plain language, without anyone feeling foolish.

Diagnosed and repaired laptop hardware

Diagnosed and repaired laptop hardware — screens, hinges, keyboards, trackpads, motherboards and power — and resolved software issues across Linux, Windows and Mac.

Recovered data from SD cards, HDDs, SSDs and RAID

Performed data recovery across a wide range of media — SD cards, HDDs, SSDs, RAID arrays, external drives and Mac systems.

Ran 24/7 support for ~1,000 IPTV/OTT servers

Provided round-the-clock 24/7 infrastructure support for an IPTV/OTT streaming platform, administering ~1,000 servers plus client-owned systems for global customers in China, the US and Germany.

See all 7 achievements

{ IT Support & Hardware }

Rescuing files from a dead machine

Hardware Repair & Data Recovery

Repairing what is worth repairing, and getting the photographs, invoices or drawings off a disk that has stopped working.

Diagnosed and repaired laptop hardware

Diagnosed and repaired laptop hardware — screens, hinges, keyboards, trackpads, motherboards and power — and resolved software issues across Linux, Windows and Mac.

Recovered data from SD cards, HDDs, SSDs and RAID

Performed data recovery across a wide range of media — SD cards, HDDs, SSDs, RAID arrays, external drives and Mac systems.

Managed 4,000 computer repairs

Managed 4,000 computer repairs, ensuring rapid and effective resolution of hardware and software issues.

See all 4 achievements

{ Brand & Marketing }

Being found and looking the part

Brand, Marketing & SEO

A name, a look and a website that turn up when somebody searches for what you sell, and read the same everywhere you appear.

Built the visual identity from 2 drawings

Built the company's icon and favicon sets from two hand-made drawings, with every derived asset regenerated by script and a check that fails when a derived file was committed before its source.

Built a 48-token macOS design system

Built a 48-token design system on the platform's own glass material and wrote the linter that rejects a magic number, a hardcoded font size, or an animation that ignores the reduce-motion preference.

Fixed a sitemap with one shared timestamp

Fixed a sitemap where 172 of 176 URLs shared one modification timestamp, by taking the date from git history after establishing that the export rewrites every file on every run.

See all 11 achievements

Tell us what you are building.

No form, no funnel, no sales sequence — just one e-mail to a person who will read it.

Start the conversation

On the server, we work with **open-source Linux** — no licences to pay for, and nothing hidden.

Ubuntu Debian

How we run servers

Choose how the work gets done

Choose an engineering partner by how the work will be done. We connect the people, decisions and delivery your project needs.

Explore our services Get in touch →

Ask concrete questions about experience

Read the work and the ways of working alongside “Full introduction”. They give you concrete questions to ask about experience and responsibility.

Cases Get in touch →

Make responsibilities clear

Start by naming the outcome and the decisions still open. We can identify where our involvement would add value and what your team should retain.

See what it costs Get in touch →

Find a way to work together

Tell us who needs the result and who will own it afterwards. We can discuss a way of working that fits your organisation.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/overview/